In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
import matplotlib.pyplot as plt
```

In [4]:
```
x=np.array([95,85,80,70,60])
```

In [5]:
```
y=np.array([85,95,70,65,70])
```

In [7]:
```
model= np.polyfit(x, y, 1)
```

In [9]:
```
model
```

Out[9]:
```
array([  0.64383562,  26.78082192])
```

In [10]:
```
predict = np.poly1d(model)
```

In [11]:
```
predict(65)
```

Out[11]:
```
68.630136986301366
```

In [12]:
```
y_pred =predict(x)
```

In [13]:
```
y_pred
```

Out[13]:
```
array([ 87.94520548,  81.50684932,  78.28767123,  71.84931507,  6
5.4109589 ])
```

In [17]:
```
from sklearn.metrics import r2_score
```

In [18]:
```
r2_score(y, y_pred)
```

Out[18]:
```
0.48032180908893263
```

In [20]:
```
y_line = model[1] + model[0]* x
```

In [21]:
```
plt.plot(x, y_line, c = 'r')
```

Out[21]:
```
[<matplotlib.lines.Line2D at 0x7fc4e8fac250>]
```

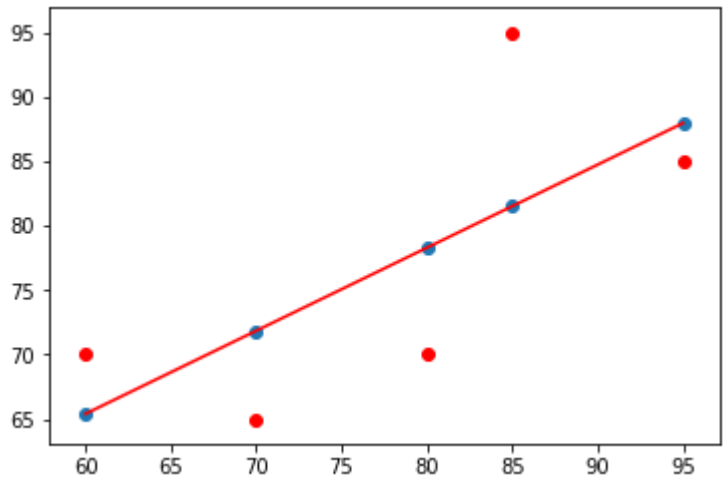
In [22]:
```
plt.scatter(x, y_pred)
```

Out[22]:
```
<matplotlib.collections.PathCollection at 0x7fc4e8fac790>
```

In [23]:
```
plt.scatter(x,y,c='r')
```

Out[23]:
```
<matplotlib.collections.PathCollection at 0x7fc4e8facc90>
```

In [24]:
```
plt.show()
```



In [25]:
```
from sklearn.datasets import load_boston
```

In [26]:
```
boston = load_boston()
```

In [27]:
```
data = pd.DataFrame(boston.data)
```

In [28]:
```
data.columns = boston.feature_names
```

In [29]:
```
data.head()
```

Out[29]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396

In [30]:
```
data['PRICE'] = boston.target
```

In [31]:
```
data.isnull().sum()
```

Out[31]:
```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
PRICE      0
dtype: int64
```

In [32]:
```
x= data.drop(['PRICE'],axis=1)
```

In [33]:
```
y=data["PRICE"]
```

In [34]:
```
from sklearn.model_selection import train_test_split
```

In [35]:
```
xtrain, xtest, ytrain, ytest =train_test_split(x, y, test_size =0.
2,random_state = 0)
```

In [36]:
```
import sklearn
```

In [37]:
```
from sklearn.linear_model import LinearRegression
```

In [38]:
```
lm = LinearRegression()
```

In [39]:
```
model=lm.fit(xtrain, ytrain)
```

In [40]:
```
ytrain_pred = lm.predict(xtrain)
```

In [41]:
```
ytest_pred = lm.predict(xtest)
```

In [42]:
```
df=pd.DataFrame(ytrain_pred,ytrain)
```

In [43]:
```
df=pd.DataFrame(ytest_pred,ytest)
```

In [44]:
```
from sklearn.metrics import mean_squared_error, r2_score
```

In [45]:
```
mse = mean_squared_error(ytest, ytest_pred)
```

In [46]:
```
print(mse)
```

```
33.4507089677
```

In [47]:
```
mse = mean_squared_error(ytrain_pred,ytrain)
```

In [48]:
```
print(mse)
```

```
19.3300193573
```

In [49]:
```
mse = mean_squared_error(ytest, ytest_pred)
```

In [50]:
```
print(mse)
```

```
33.4507089677
```

In [53]:
```
plt.scatter(ytrain ,ytrain_pred,c='blue',marker='o',label='Trainin
g data')
```

Out[53]:
```
<matplotlib.collections.PathCollection at 0x7fc4e63c7310>
```

In [54]:
```
plt.scatter(ytest,ytest_pred ,c='lightgreen',marker='s',label='Tes
t data')
```

Out[54]:
```
<matplotlib.collections.PathCollection at 0x7fc4e63c7910>
```

In [55]:
```
plt.xlabel('True values')
```

Out[55]:
```
Text(0.5,0,u'True values')
```

In [56]:
```
plt.ylabel('Predicted')
```

Out[56]:
```
Text(0,0.5,u'Predicted')
```

In [57]:
```
plt.title("True value vs Predicted value")
```

Out[57]:
```
Text(0.5,1,u'True value vs Predicted value')
```

In [58]:
```
plt.legend(loc= 'upper left')
```

Out[58]:
```
<matplotlib.legend.Legend at 0x7fc4e573f490>
```

In [59]:
```
plt.legend(loc= 'upper left')
```

Out[59]:
```
<matplotlib.legend.Legend at 0x7fc4e63c7f90>
```

In [60]:
```
plt.show()
```